

Práctica 3 SWAP

XuSheng Zheng

Índice

1. Creación de la tercera máquina	2
2. Nginx	2
2.1. Opciones avanzadas	4
3. Haproxy	6
3.1. Opciones avanzadas	7
4. Bibliografía	9

1. Creación de la tercera máquina

Para esta práctica, necesitamos crear una tercera máquina virtual que nos servirá de balanceador. Para ello, especificamos los siguientes datos siguiendo los mismos pasos que en la práctica 1:

Una vez instalada la máquina, necesitamos configurar la conexión a red. Lo hacemos de la misma manera que en la práctica 1:

```
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.56.72/24]
  version: 2
```

Y comprobamos con `ifconfig`:

```
xuzheng@m3-xuzheng:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe83:c3e2 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:83:c3:e2 txqueuelen 1000 (Ethernet)
    RX packets 84 bytes 34018 (34.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 89 bytes 10971 (10.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.72 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fee2:7a5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e2:07:a5 txqueuelen 1000 (Ethernet)
    RX packets 694 bytes 211816 (211.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 2610 (2.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 100 bytes 7900 (7.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 100 bytes 7900 (7.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2. Nginx

Instalamos **nginx** en m3 mediante **apt-get** con los comandos del guion y comprobamos que está activo:

```
kuzheng@m3-xuzheng:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-04-05 15:20:39 UTC; 45s ago
     Docs: man:nginx(8)
   Main PID: 15276 (nginx)
      Tasks: 2 (limit: 4653)
   CGroup: /system.slice/nginx.service
           └─15276 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
              15278 nginx: worker process

abr 05 15:20:39 m3-xuzheng systemd[1]: Starting A high performance web server and a reverse proxy se
abr 05 15:20:39 m3-xuzheng systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid:
abr 05 15:20:39 m3-xuzheng systemd[1]: Started A high performance web server and a reverse proxy se
```

Ahora procedemos a la configuración: en primer lugar deshabilitamos la funcionalidad de servidor web editando en `/etc/nginx/nginx.conf` comentando la línea `include /etc/nginx/sites-enabled/*;`:

```
##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
#include /etc/nginx/sites-enabled/*;
```

Creamos el archivo de configuración `/etc/nginx/conf.d/default.conf` con los siguientes datos:

```
upstream balanceo_xuzheng{
    server 192.168.56.70;
    server 192.168.56.71;
}

server{
    listen 80;
    server_name balanceador_xuzheng;

    access_log /var/log/nginx/balanceador_xuzheng.access.log;
    error_log /var/log/nginx/balanceador_xuzheng.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_xuzheng;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

Una vez guardado, reiniciamos el servicio **nginx**. Para comprobar el funcionamiento del balanceador, se ha modificado el archivo `/var/www/html/swap.html` para que se pueda distinguir las máquinas:

```
xuzheng@m1-xuzheng:~$ cat /var/www/html/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>
```

```
xuzheng@m2-xuzheng:~$ cat /var/www/html/swap.html
<HTML>
<BODY>
SWAP M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>
```

Ahora podemos comprobar el funcionamiento de **nginx** usando **cURL** desde el anfitrión:

```

xdi@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP_M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

xdi@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP_M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

```

En este caso hemos tratado las dos máquinas por iguales. Podemos dar más peso a la primera máquina mediante el modificador **weight**:

```

upstream balanceo_xuzheng{
    server 192.168.56.70 weight=2;
    server 192.168.56.71 weight=1;
}

server{
    listen 80;
    server_name balanceador_xuzheng;

    access_log /var/log/nginx/balanceador_xuzheng.access.log;
    error_log /var/log/nginx/balanceador_xuzheng.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_xuzheng;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}

```

Ahora de cada 3 peticiones, 2 serán atendidas por m1 y 1 por m2:

```

xdi@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP_M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

xdi@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP_M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

xdi@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP_M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

```

2.1. Opciones avanzadas

Para evitar el conflicto de estados y sesiones entre los servidores, **nginx** permite dirigir las peticiones provenientes de un determinado IP al mismo servidor final mediante la opción **ip_hash**.

La directiva **keepalive** determina el número máximo de conexiones a servidores finales que se mantiene en

caché y **keepalive_requests** determina el número máximo de peticiones que pueden servir a través de estas conexiones, una vez que el número de peticiones exceda el máximo, la conexión se cerrará.

```

upstream balanceo_xuzheng{
    ip_hash;
    server 192.168.56.70 weight=2;
    server 192.168.56.71 weight=1;
    keepalive 3;
}

server{
    listen 80;
    server_name balanceador_xuzheng;

    access_log /var/log/nginx/balanceador_xuzheng.access.log;
    error_log /var/log/nginx/balanceador_xuzheng.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_xuzheng;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}

```

Comprobamos el funcionamiento a través del anfitrión:

```

root@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP_M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

root@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP_M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

root@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP_M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

root@DESKTOP-S58Q8SA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP_M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

```

Podemos que en este caso sólo sirve m2 pues estamos enviando a través del mismo IP.

Para gestionar caídas de los servidores podemos usar los comandos **max_fails** para especificar el número máximo de intentos de comunicación fallidos antes de considerar al servidor no operativo y **fail_timeout** para especificar el periodo de tiempo con el que se considera esos intentos. En el siguiente ejemplo consideramos que el servidor no será operativo si existen 3 intentos fallidos en un periodo de 30 segundos:

```

upstream balanceo_xuzheng{
    ip_hash;
    server 192.168.56.70 weight=2 max_fails=3 fail_timeout=30s;
    server 192.168.56.71 weight=1 max_fails=3 fail_timeout=30s;
    keepalive 3;
}

server{
    listen 80;
    server_name balanceador_xuzheng;

    access_log /var/log/nginx/balanceador_xuzheng.access.log;
    error_log /var/log/nginx/balanceador_xuzheng.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_xuzheng;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}

```

3. Haproxy

Para este apartado, paramos y desactivamos **nginx** para evitar conflictos e instalamos **haproxy** mediante **apt-get**.

```

kuzheng@m3-kuzheng:~$ sudo systemctl disable nginx.service
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install
Executing: /lib/systemd/systemd-sysv-install disable nginx
kuzheng@m3-kuzheng:~$ sudo systemctl stop nginx.service
kuzheng@m3-kuzheng:~$ sudo systemctl status nginx.service
• nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:nginx(8)

abr 07 09:02:08 m3-kuzheng systemd[1]: Starting A high performance web server and a reverse proxy se
abr 07 09:02:08 m3-kuzheng systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid:
abr 07 09:02:08 m3-kuzheng systemd[1]: Started A high performance web server and a reverse proxy se
abr 07 09:12:49 m3-kuzheng systemd[1]: Stopping A high performance web server and a reverse proxy se
abr 07 09:12:49 m3-kuzheng systemd[1]: Started A high performance web server and a reverse proxy se
abr 07 09:12:49 m3-kuzheng systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid:
abr 07 09:12:49 m3-kuzheng systemd[1]: Started A high performance web server and a reverse proxy se
abr 07 09:16:20 m3-kuzheng systemd[1]: Stopping A high performance web server and a reverse proxy se
abr 07 09:16:20 m3-kuzheng systemd[1]: Stopped A high performance web server and a reverse proxy se
kuzheng@m3-kuzheng:~$ sudo apt-get install haproxy_

```

Comprobamos que está activo:

```

kuzheng@m3-kuzheng:~$ sudo systemctl status haproxy.service
• haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-04-07 09:17:36 UTC; 1min 25s ago
     Docs: man:haproxy(1)
           file:/usr/share/doc/haproxy/configuration.txt.gz
   Main PID: 2720 (haproxy)
    Tasks: 2 (limit: 4653)
   CGroup: /system.slice/haproxy.service
           └─2720 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid
           └─2721 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid

abr 07 09:17:36 m3-kuzheng systemd[1]: Starting HAProxy Load Balancer...
abr 07 09:17:36 m3-kuzheng systemd[1]: Started HAProxy Load Balancer.

```

Pasamos ahora a configurar **haproxy**: añadimos al archivo `/etc/haproxy/haproxy.cfg` las siguientes líneas para tener una configuración round-robin básica:

```

frontend http-in
    bind *:80
    default_backend balanceo_xuzheng

backend balanceo_xuzheng
    server m1 192.168.56.70:80 maxconn 32
    server m2 192.168.56.71:80 maxconn 32

```

Reiniciamos el servicio y comprobamos desde el anfitrión:

```
xdl6@DESKTOP-SS3Q82A WINE64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

xdl6@DESKTOP-SS3Q82A WINE64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

xdl6@DESKTOP-SS3Q82A WINE64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

xdl6@DESKTOP-SS3Q82A WINE64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>
```

Para dar más peso a determinados servidores, podemos usar la opción **weight** con números entre 0 y 256 en las líneas de **server**. Para tener la analogía con el apartado de **nginx**, asignamos a m1 el doble de carga que m2:

```
frontend http-in
    bind *:80
    default_backend balanceo_xuzheng

backend balanceo_xuzheng
    server m1 192.168.56.70:80 weight 20 maxconn 32
    server m2 192.168.56.71:80 weight 10 maxconn 32
```

Y comprobamos:

```
xdl6@DESKTOP-SS3Q82A WINE64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

xdl6@DESKTOP-SS3Q82A WINE64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

xdl6@DESKTOP-SS3Q82A WINE64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M2
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

xdl6@DESKTOP-SS3Q82A WINE64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>
```

Vemos que efectivamente m1 atiende 2 de las 3 peticiones.

3.1. Opciones avanzadas

Al igual que **nginx**, **haproxy** también permite el balanceo por IP mediante la opción **hash-type consistent**.

Además, **haproxy** permite 3 tipos de comprobaciones sobre el estado de los servidores:

- Activo: por defecto en este caso **haproxy** intentará establecer una conexión TCP con el servidor final cada 2 segundos. Tras 3 conexiones fallidas se considerará que el servidor está caído y no se le enviará peticiones hasta que consiga 2 conexiones exitosas.
- Pasivo: similar a la opción **keepalive_requests** de **nginx**, establece un límite de errores consecutivos que puede haber antes de dar por caído al servidor.
- Agente: mediante un agente externo del servidor final, **haproxy** puede controlar el estado con el que se encuentra el servidor.

En el siguiente ejemplo establecemos un máximo de 10 errores:

```
frontend http-in
  bind *:80
  default_backend balanceo_xuzheng

backend balanceo_xuzheng
  balance source
  hash-type consistent
  server m1 192.168.56.70:80 weight 20 maxconn 32 check observe layer7 error-limit 10 on-error
  mark-down
  server m2 192.168.56.71:80 weight 10 maxconn 32 check observe layer7 error-limit 10 on-error
  mark-down
```

Con **observe layer7** indicamos que se consideren todas las respuestas HTTP del servidor y con **on-error mark-down** indicamos que el servidor está caído cuando se alcance los 10 errores. Para comprobar el funcionamiento del balanceo por IP, mandamos peticiones desde el anfitrión:

```
root@DESKTOP-SS3QSCA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

root@DESKTOP-SS3QSCA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>

root@DESKTOP-SS3QSCA MINGW64 /e/DGIIM/QUINTO 2º CUAT/SWAP/Prácticas/P3 (main)
$ curl http://192.168.56.72/swap.html
<HTML>
<BODY>
SWAP M1
Web de ejemplo de xuzheng para SWAP
Email:xuzheng@correo.ugr.es
</BODY>
</HTML>
```

Podemos apreciar que todas las peticiones son atendidas por m1 pues provienen del mismo IP.

4. Bibliografía

- http://nginx.org/en/docs/http/nginx_http_upstream_module.html
- <http://www.haproxy.org/download/1.4/doc/configuration.txt>
- <https://serverfault.com/questions/113637/haproxy-roundrobin-weights>
- <https://www.haproxy.com/blog/client-ip-persistence-or-source-ip-hash-load-balancing>
- <https://www.haproxy.com/blog/how-to-enable-health-checks-in-haproxy>
- <https://serverfault.com/questions/123629/run-task-every-90-minutes-with-cron>